

***C# BASICS***

**Training Assignments**

|  |  |
| --- | --- |
| Document Code | 25e-BM/HR/HDCV/FSOFT |
| Version | 1.1 |
| Effective Date | 20/11/2012 |

**Hanoi, 06/2019**

RECORD OF CHANGES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Effective Date | Change Description | Reason | Reviewer | Approver |
|  | 01/Oct/2018 | Create new | Draft |  |  |
|  | 01/Jun/2019 | Update template | Fsoft template | DieuNT1 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Contents

[Day 1: Assignment 3 : Operators 4](#_Toc12007474)

[Objectives: 4](#_Toc12007475)

[Business needs: 4](#_Toc12007476)

[Prerequisites: 4](#_Toc12007477)

[Technologies: 4](#_Toc12007478)

[1. Basic Assignment Operator 5](#_Toc12007479)

[2. Arithmetic Operators 5](#_Toc12007480)

[3. Relational Operators 5](#_Toc12007481)

[4. Logical Operators 6](#_Toc12007482)

[5. Unary Operators 6](#_Toc12007483)

[6. Ternary Operator 7](#_Toc12007484)

[7. Bitwise and Bit Shift Operators 8](#_Toc12007485)

[8. Compound Assignment Operators 9](#_Toc12007486)

|  |  |
| --- | --- |
|  | **CODE: Net.S.A003**  **TYPE: SHORT**  **LOC: 100**  **DURATION: 60 MINUTES** |

# Day 1: Assignment 3 : Operators

**Objectives:**

* Understand and practice with Basic Operator in C#.
* Understand and practice with Arithmetic Operators; Relational Operators; Logical Operators; Unary Operators; Ternary Operator; Bitwise and Bit Shift Operators; Compound Assignment Operators.

**Business needs:**

* Create a Window Console application bases on Operator in C#.

**Prerequisites:**

* Working environment: Visual Studio 2013 or higher.
* Delivery: Source code packaged in a compress archive.

**Technologies:**

The product implements one or more technology:

* Operator
* Arithmetic Operators
* Logical Operators
* Unary Operators
* Ternary Operator
* Bitwise and Bit Shift Operators
* Compound Assignment Operators

1. Basic Assignment Operator

Basic assignment operator (=) is used to assign values to variables.

double x;

x = 50.05;

string name;

name = “Peter”;

string welcome Text = “Hello:” + name;

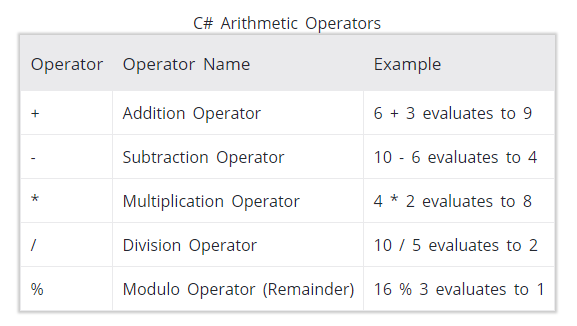
1. Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations such as addition, subtraction, multiplication, division, etc.

int x = 5;

int y = 10;

int z = x + y;// z = 15



1. Relational Operators

int firstNumber = 10, secondNumber = 20;

bool result1 = (firstNumber == secondNumber); //// false

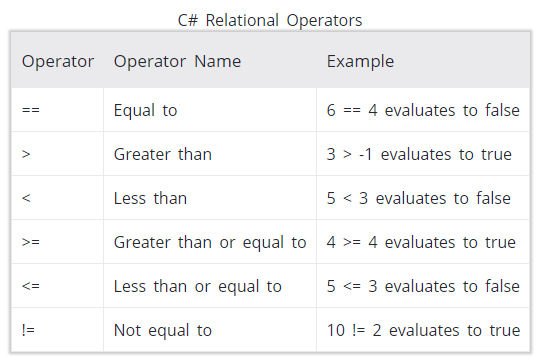
bool result2 = (firstNumber > secondNumber); //// false

bool result3 = (firstNumber < secondNumber); //// true

bool result4 = (firstNumber >= secondNumber); //// false

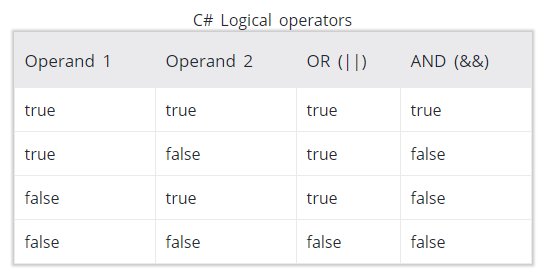
bool result5 = (firstNumber <= secondNumber); //// true

bool result6 = (firstNumber != secondNumber); //// true



1. Logical Operators

Logical operators are used to perform logical operation such as and, or. Logical operators operates on boolean expressions (true and false) and returns boolean values. Logical operators are used in decision making and loops.



1. Unary Operators

Unary operators operates on a single operand.

int number = 10, result;

bool flag = true;

result = +number;

Console.WriteLine("+number = " + result);

result = -number;

Console.WriteLine("-number = " + result);

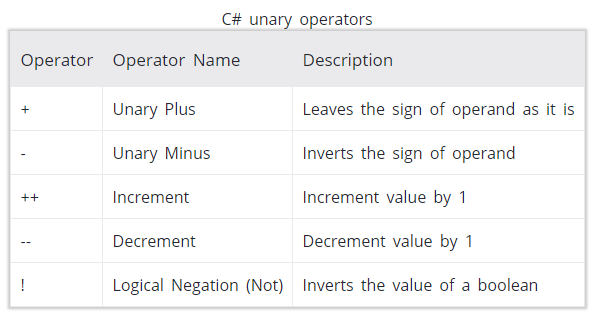
result = ++number;

Console.WriteLine("++number = " + result);

result = --number;

Console.WriteLine("--number = " + result);

Console.WriteLine("!flag = " + (!flag));



1. Ternary Operator

The ternary operator ? : operates on three operands. It is a shorthand for if-then-else statement. Ternary operator can be used as follows:

variable = Condition? Expression1 : Expression2;

The ternary operator works as follows:

* If the expression stated by Condition is true, the result of Expression1 is assigned to variable.
* If it is false, the result of Expression2 is assigned to variable.

int number = 10;

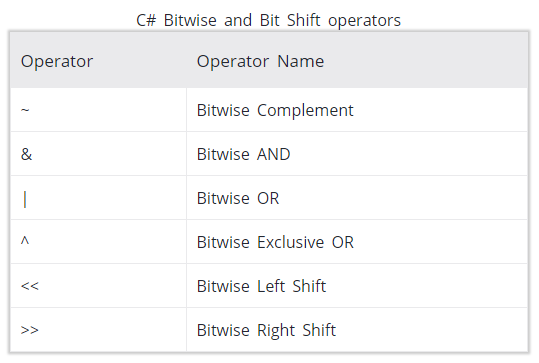
string result;

result = (number % 2 == 0)? "Even Number" : "Odd Number";

Console.WriteLine("{0} is {1}", number, result);

1. Bitwise and Bit Shift Operators

Bitwise and bit shift operators are used to perform bit manipulation operations.



Run the code and see results:

int firstNumber = 10;

int secondNumber = 20;

int result;

result = ~firstNumber;

Console.WriteLine("~{0} = {1}", firstNumber, result);

result = firstNumber & secondNumber;

Console.WriteLine("{0} & {1} = {2}", firstNumber,secondNumber, result);

result = firstNumber | secondNumber;

Console.WriteLine("{0} | {1} = {2}", firstNumber,secondNumber, result);

result = firstNumber ^ secondNumber;

Console.WriteLine("{0} ^ {1} = {2}", firstNumber,secondNumber, result);

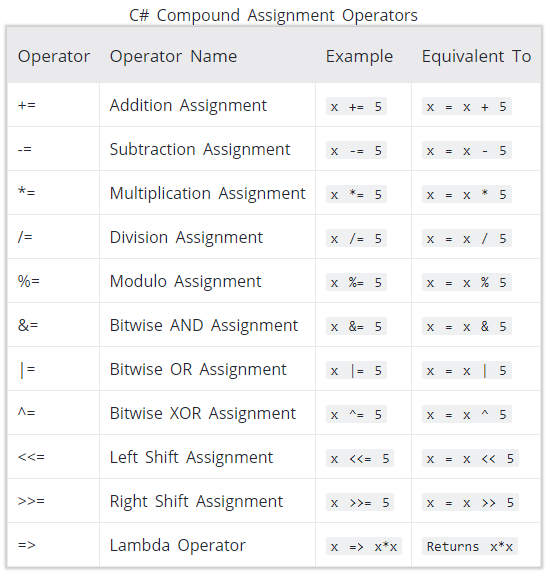
result = firstNumber << 2;

Console.WriteLine("{0} << 2 = {1}", firstNumber, result);

result = firstNumber >> 2;

Console.WriteLine("{0} >> 2 = {1}", firstNumber, result);

1. Compound Assignment Operators



Run the code and see results:

int number = 10;

number += 5;

Console.WriteLine(number);

number -= 3;

Console.WriteLine(number);

number \*= 2;

Console.WriteLine(number);

number /= 3;

Console.WriteLine(number);

number %= 3;

Console.WriteLine(number);

number &= 10;

Console.WriteLine(number);

number |= 14;

Console.WriteLine(number);

number ^= 12;

Console.WriteLine(number);

number <<= 2;

Console.WriteLine(number);

number >>= 3;

Console.WriteLine(number);